

Amendments to the Claims:

1. (Currently Amended) A method comprising
~~getting an object header from an object~~ performing a first type checking on an object
to find out a hotspot in the first type checking, and
performing a second type checking between a class of the object and a target class
specified by the hotspot to assert an indicator in an object header of the object to indicate
~~checking from the object header~~ a success result of a first time the second type checking at the
hotspot between a class of the object and a target class specified by a hotspot in the first time
type checking.
2. (Canceled)
3. (Currently Amended) The method of claim 1 further comprising
~~determining whether the object header comprises an indicator that is deasserted~~
~~deasserting the indicator~~ to indicate a failure of the second first time type checking ~~failure~~
between the object class and the target class ~~associated with the indicator~~.
4. (Currently Amended) The method of claim 1 further comprising
skipping a ~~second time~~ third type checking between the object class and the target
class at the hotspot, in response to determining that the indicator of the object header
~~comprises an indicator that~~ is asserted ~~to indicate a first time type checking success~~.
5. (Currently Amended) The method of claim 1 further comprising
~~performing~~ skipping a ~~second time~~ third type checking between the object class and
the target class, in response to determining that the indicator of the object header ~~comprises an~~
~~indicator that~~ is deasserted to indicate a failure of the second first time type checking ~~failure~~.
6. (Currently Amended) The method of claim 1 further comprising
detecting the hotspot in the first ~~time~~ type checking by dynamic profiling.

7. (Currently Amended) A system, comprising

a processor to ~~get an object header from an object, and obtain from the object header a result of a first time type checking at~~ find out a hotspot in a first type checking between for a class of the an object, and performing a second type checking between the object class and a target class specified by the hotspot to indicate by an indicator in a header of the object a result of the second type checking at the hotspot; and

a memory to save the target class.

8. (Currently Amended) The system of claim 7, wherein the processor further to

determine whether a third that the first time type checking between the object class and the target class is successful at the hotspot based on a logic value of is successful, in response to detecting that the object header comprises an indicator associated with the target class that has a first logic value.

9. (Currently Amended) The system of claim 7, wherein the processor further to

~~perform~~ skipping a second time third type checking between on the object at the hotspot class and the target class, in response to detecting that an the indicator associated with the target class in the object header has a second logic value to indicate a failure of the second type checking.

10. (Currently Amended) The system of claim 7, wherein the processor further to

traverse a class hierarchy associated with the class of the object for the second type chekcing, ~~in response to determining that the first time type checking at the hotspot is failed.~~

11. (Currently Amended) The system of claim 7, wherein the processor further to

assert ~~an the indicator associated with the target class in the object header,~~ in response to determining in a the second time type checking at the hotspot that the class of the object and the target class match a type checking condition.

12. (Currently Amended) The system of claim 7, wherein the processor further to return a signal indicating that the second type checking is successful, in response to determining that the class of the object and the target class match a predetermined criterion.

13. (Currently Amended) The system of claim 7, wherein the memory further to save a beginning address of a handle of the target class, and wherein the processor further to ~~detecting~~ detect the hotspot by dynamic profiling.

14. (Currently Amended) A tangible machine readable storage medium comprising a plurality of instructions that in response to being executed result in a computing device

~~obtaining an object header from~~ find out a hotspot in a first type checking for a class of an object , and

performing a second type checking ~~between the object class checking a bit indicator in the object header to indicate a result of a first time type checking at a hotspot between a~~ the class of the object and a target class specified by the hotspot to indicate by a bit indicator in a header of the object a result of the second type checking at the hotspot .

15. (Currently Amended) The tangible machine readable storage medium of claim 14, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device

skipping a ~~second time~~ third type checking at the hotspot between the object class and the target class, in response to determining that the bit indicator is asserted to indicate a successful result of the second type checking.

16. (Currently Amended) The tangible machine readable storage medium of claim 14, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device

~~performing~~ skipping a ~~second time~~ third type checking at the hotspot between the object class and the target class, in response to determining that the bit indicator is deasserted to indicate a failure result of the second type checking.

17. (Currently Amended) The tangible machine readable storage medium of claim 14, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device

detecting at the hotspot whether the object class and the target class match a type checking criterion, ~~in response to determining that~~ based on the bit indicator ~~indicates a failure result.~~

18. (Currently Amended) The tangible machine readable storage medium of claim 16, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device

asserting the bit indicator, in response to determining in the second ~~time~~ type checking that the object class and the target class match a type checking criterion.

19. (Currently Amended) The tangible machine readable storage medium of claim 16, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device

at the hotspot, returning a signal indicating whether the object class and the target class match a type checking criterion based on the bit indicator ~~a result of the second time type checking.~~

20. (Currently Amended) The tangible machine readable storage medium of claim 16, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device

throwing an exception at the hotspot, in response to determining from the bit indicator that the second ~~time~~ type checking for the hotspot is failed~~successful~~.

21. (Currently Amended) The tangible machine readable storage medium of claim 16, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device

pushing a result code on a stack to indicate whether ~~the~~ a type checking for the hotspot is successful based on the bit indicator.

22. (Currently Amended) A system comprising,

a processor,

a memory coupled to the processor, wherein the memory is to store

a compiler to convert source code associated with a first time type checking for an object into byte code;

a loader coupled with the compiler to load the byte code;

a dynamic compiler coupled with the loader to receive the byte code from the loader, and to generate first native code associated with the first time type checking based on the byte code; and

a profiler coupled with the dynamic compiler to detect a hotspot in the first time type checking based on the first native code, and to return the hotspot to the dynamic compiler.

23. (Currently Amended) The system of claim 22, wherein the dynamic compiler further to regenerate second native code that calls a second type checking ~~function~~ for the hotspot to determine a type checking result between a class of the object and a target class specified by the hotspot ~~from an object header of the object~~.

24. (Currently Amended) The system of claim ~~22~~ 23, wherein the dynamic compiler further to assert an indicator in an object header of the object to a first logic value to indicate regenerate ~~second native code that calls a type checking function for the hotspot to detect a~~ success of the second type checking ~~success~~ between a class of the object and a target class specified by the hotspot, ~~in response to determining that an indicator associated with the target class in an object header of the object has a first logic value.~~

25. (Currently Amended) The system of claim ~~22~~ 23, wherein the dynamic compiler further to regenerate second native code that calls a second type checking ~~function~~ for the hotspot to ~~detect~~ indicate a failure of the second type checking ~~failure~~ between a class of the object and a

target class specified by the hotspot, ~~in response to determining that an indicator associated with the target class in an object header of the object has~~ by a second logic value in a header of the object.

26. (Currently Amended) The system of claim ~~22~~ 23, wherein the dynamic compiler further to regenerate second native code that calls a type checking function for the hotspot to skip type checking between a class of the object and a target class specified by the hotspot, in response to determining that an object header of the object indicates a success of the second ~~a~~ type check ~~success~~.

27. (Currently Amended) The system of claim ~~22~~ 23, wherein the dynamic compiler further to regenerate second native code that calls a type checking function for the hotspot to ~~perform~~ skip type checking between a class of the object and a target class specified by the hotspot, in response to determining that an indicator associated with the target class in an object header of the object is deasserted in the second type checking.

28. (Currently Amended) The system of claim ~~22~~ 23, wherein the dynamic compiler further to regenerate second native code that calls a type checking function for the hotspot to assert an indicator in an object header of the object, in response to a success of the second type checking ~~success~~ between a class of the object and a target class specified by the hotspot.

29. (Currently Amended) The system of claim 22, wherein the dynamic compiler further to regenerate second native code that calls a second type checking ~~function~~ for the hotspot to traverse super classes of a class of the object to determine whether one of the super classes is the same as a target class as specified by the hotspot, ~~in response to determining a type checking failure between the class of the object and the target class from an object header of the object~~.

30. (Canceled).